Gesture Recognition Assignment

Problem Statement:-

As a data scientist at a home electronics company which manufactures state of the art smart televisions. We want to develop a cool feature in the smart-TV that can recognize five different gestures performed by the user which will help users control the TV without using a remote.

- Thumbs up : Increase the volume.
- Thumbs down : Decrease the volume.
- Left swipe : 'Jump' backwards 10 seconds.
- Right swipe : 'Jump' forward 10 seconds.
- Stop : Pause the movie.

Observations:-

MODEL	HYPERPARAMETER	RESULTS	DECISION + EXPLANATION
MODEL-1 Conv3D	**Total params:** 8,317,701 **Trainable params:** 8,314,757 **Non-trainable params:** 2,944	Epochs: 50 Batch_size=32 Train accuracy: 0.99 Validation accuracy: 0.51	The model is clearly over fitting as the training and validation accuracy have a huge difference ie…over fitting, re-introducing Dropout layers.
MODEL-2 CONV30 + Dropout layer	**Total params:** 8,317,701 **Trainable params:** 8,314,757 **Non-trainable params:** 2,944	Epochs: 50 Batch_size=32 Train accuracy: 0.97 Validation accuracy: 0.56	After using the dropouts of 0.2 and 0.5 after each layer we can still see that the data is over fitting.so now we try to implement a new architecture. And the accuracy is reduced from 99% to 97%.
MODEL-3 Time distributed + Dense layer	**Total params:** 129,477 **Trainable params:** 128,517 **Non-trainable params:** 960	Epochs: 50 Batch_size=32 Train accuracy: 0.99 Validation accuracy: 0.59	Using the time distributed and dense layer architecture the trainable and non-trainable parameters have decreased but it had a good training accuracy and not validation so model over fits we will use diff architecture.

MODEL-4 Time distributed + GRU	**Total params:** 99,845 **Trainable params:** 99, 269 **Non-trainable params:** 576	Epochs: 50 Batch_size=32 Train accuracy:0.98 Validation accuracy:0.56	So when compared to above architecture we can observe that the total no.of parameters have been reduced though the accuracy is slightly different and which is over fitting too so lets try using LSTM architecture
FINAL MODEL time Distributed + LSTM2D	**Total params:** 13,781 **Trainable params:** 13,589 **Non-trainable params:** 192	Epochs: 50 Batch_size=8 Train accuracy:0.92 Validation accuracy: 0.93	Choosing the time dense+LSTM is giving us a good accuracy of about 92% of train and 93% on validation hence its not over fitting and we can consider it as our final model.

Conclusion:-

Time Distributed +LSTM model gave a training accuracy of 92% and validation accuracy of 93% which clearly shows that model isn't over fitting and hence we take out final model to be Time Distributed+LSTM.